

EL764005921

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Detecting Multiple Objects in Digital Image Data

Inventor:

Cormac Herley

ATTORNEY'S DOCKET NO. MS1-719US

1 **TECHNICAL FIELD**

2 The following description relates to image processing. More particularly,
3 the following description relates to using image-processing techniques to detect
4 multiple objects in digital image data.

5

6 **BACKGROUND**

7 Scanners are often used to scan objects such as photographs. These
8 scanned objects can then be used for many different purposes such as
9 incorporation in a Web page, sending to friends and family over electronic mail,
10 and/or the like. Original Equipment Manufacturers (OEMs) and Independent
11 Hardware Vendors (IHVs) typically distribute scanners with image-processing
12 software to simplify a user's scanning experience. Image processing software is
13 typically configured to automatically analyze scanned image data to detect a single
14 scanned object of interest. Upon detecting the single object, the image processing
15 software typically either highlights the detected object with an outline and/or
16 automatically segments the detected object from the scanned image data.

17 To speed up the scanning process when multiple objects such as
18 photographs need to be scanned, a user will typically place two, three, or more
19 objects at a time onto the scanner's platen. After the multiple objects have been
20 scanned, image processing software, rather than detecting multiple objects in the
21 scanned image data, only typically detects a single large object comprised of the
22 smaller multiple objects. Thus, the user generally needs to manually segment the
23 single large object into several smaller objects to fully represent the multiple
24 objects that were placed on the scanner's platen. Such manual segmenting is
25 typically performed through the use of manual segmenting tools that are

1 distributed with the image processing software. This process can be tedious and
2 timing consuming for the user.

3 Accordingly, the invention arose out of concerns associated with providing
4 improved systems and methods for scanning objects.

5

6 **SUMMARY**

7 The described subject matter automatically detects the boundaries of
8 multiple objects in digital image data such as scanned image data. A background
9 color such as a scanner lid background color is detected. An edge map is
10 generated based on a set of transitions between the determined background color
11 and other colors that correspond to the scanned image data. A set of straight lines,
12 or boundaries that correspond to one or more objects in the image data is
13 determined by analyzing the generated edge map. The straight lines are assigned
14 to respective objects based on a set of rules. Thus, scanned image data is
15 automatically segmented based on the detected boundaries into separate image
16 objects—thus, sparing a user the trouble of carrying out the image segmentation
17 by hand.

18

19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 Fig. 1 is a block diagram that shows aspects of an exemplary system to
21 detect multiple objects in image data.

22 Fig. 2 is a graph that illustrates that peaks in Hough transform space
23 correspond to straight lines.

1 Fig. 3 is a block diagram that shows that all pixels that lie on a line need not
2 all be contiguous to one another to be categorized as belonging to the same line by
3 a Hough transform.

4 Fig. 4 is a block diagram that shows that a Hough transform will
5 erroneously detect an infinite line rather than a finite line with well defined
6 endpoints if two or more objects are aligned.

7 Fig. 5 is a block diagram that shows aspects of an exemplary computer to
8 detect multiple objects in image data.

9 Fig. 6 is a block diagram that shows aspects of an exemplary procedure to
10 detect multiple objects in image data.

11 Fig. 7 is a block diagram that shows other aspects of an exemplary
12 procedure to detect multiple objects in image data. Specifically, Fig. 7 illustrates
13 how a candidate object is detected in image data after a previous object has been
14 detected.

15

16 **DETAILED DESCRIPTION**

17 The following description sets forth a number of specific implementations
18 to detect multiple objects in digital image data. The subject matter
19 incorporates elements recited in the appended claims. The implementations are
20 described with specificity in order to meet statutory requirements. However, the
21 description itself is not intended to limit the scope of this patent. Rather, it has
22 been contemplated that the claimed subject matter might also be embodied in other
23 ways, to include different elements or combinations of elements similar to the
24 ones described in this document, in conjunction with other present or future
25 technologies.

1 **Exemplary System**

2 Fig. 1 shows an exemplary system 100 to detect boundaries of multiple
3 objects in image data. System 100 includes a computer 102 connected to a
4 peripheral device 114. The peripheral device 106 can be any type of peripheral
5 device such as a scanner, a digital camera, and the like, that is used to transfer
6 image data to the computer.

7 Computer 102 has one or more processors 104 and one or more forms of
8 computer-readable memory media 106 such as electronic memory, magnetic
9 storage media, optical storage media, or some other type of data storage.
10 Programs are stored in the memory from where they are executed by the processor.
11 In this example, such programs include a digital image object detection program
12 module 110, and well as an operating system 108 such as a Microsoft
13 WINDOWS ® operating system..

14 The digital image object detection application 110 provides various services
15 such as obtaining digital image data 112, detecting whether the image data
16 includes one or more objects, determining the boundaries of the detected object(s),
17 and optionally segmenting the detected objects based on the determined object
18 boundaries. The image data may be obtained in a number of different ways. For
19 example, image data may be communicated across a network (e.g., e-mail
20 attachments), acquired from a peripheral device 114 such as a scanner, a camera,
21 and/or the like.

22 In one implementation the peripheral 114 is a scanner. In this
23 implementation, boundaries that correspond to multiple objects in the image date,
24 the system 100 first determines an estimate of a background color of the scanner
25

1 lid. This can be accomplished by determining a pixel value near the edge of the
2 scanner platen. This can also be accomplished by forming a histogram of the
3 scanned data and seeking a dominant color.

4 Next, an edge map of the scanned image data is generated. An edge map is
5 an array of elements, wherein each element represents an indication of whether a
6 respective pixel of the image data corresponds to a potential edge in an image such
7 as a photograph. A pixel is considered to be an edge or boundary of an object if
8 the absolute difference between the pixel and the determined background color
9 exceeds a certain threshold. This threshold can be set to some percentage of the
10 dynamic range of the scanner such that a certain distribution of pixel values will
11 qualify as a background color.

12 For example, consider that a particular scanner outputs pixel values that
13 range from zero (0) to two-hundred-and-fifty-five (255) and the scanner
14 background color is determined to be equivalent to a pixel value of 255. A
15 threshold of four (4) will indicate that pixel values of two-hundred-and-fifty-
16 one (251) to 255 represent a background color. Thus, in this example, a pixel
17 value of 250 may represent an edge. The threshold can be selected such that
18 picture or scanning noise can be accounted for in the background determination.
19 In this implementation, the threshold is adjusted to take into account the difference
20 between the background color and the first pixel of the image. For color images,
21 which generally have three or more color planes such as red, green, and blue
22 channels, the edge locations can be estimated by carrying out this “thresholding”
23 along only one of the color channels, for example along a green channel

24 The generated edge map may be extremely noisy, either missing data or
25 having a number of data that do not correspond to edges. For example, if a

1 scanned photograph includes a transition from a dark horizon to a light colored
2 sky, wherein the sky's color is similar to the background color of the scanner lid,
3 the sky may be represented in the edge map as noise. Thus, the determination of
4 the existence of straight lines from an edge map is prone to substantial error.

5 A Hough transform of the edge map is taken to determine which of the
6 detected edges correspond to straight lines in the image data. (A Hough transform
7 is a well-known transform used to determine the existence of edges in digital
8 image data). This transform maps every point (x,y) in the edge map to a curve in
9 the (ρ (rho), θ (theta)) space, where rho is the perpendicular distance from some
10 origin of a line at angle theta passing through the point (x,y). Points in the Hough
11 space that are at the intersection of many such curves are points, or "peaks" that
12 correspond to straight lines in the image space. In this manner, straight lines in the
13 image space are determined by identifying those points in Hough space that
14 illustrate clustering, or peaks.

15 Fig. 2 is a graph that illustrates that a single rectangular object will produce
16 four (4) peaks in the Hough space. The vertical axis 202 represents the
17 "Rho" coordinates of a point in Hough space. The horizontal axis 204 represents
18 the "Theta" coordinates of a point in Hough space. Peaks having the same theta
19 coordinate such as peak 206 and peak 208 represent parallel lines. (Peak 210 and
20 peak 212 also share a theta coordinate indicating that they are also parallel to one
21 another).

22 A rectangular object will have a first pair of lines with corresponding theta
23 coordinates that are ninety-degrees apart from the theta coordinates that
24 correspond to a second pair of parallel lines. In this example, two parallel lines
25 respectfully represented by peak 206 and peak 208 have theta values that are

1 ninety-degrees apart from the theta coordinates that correspond to a second pair of
2 parallel lines respectfully represented by peak 210 and peak 212. In this manner, a
3 single rectangular object can be identified using a Hough transform.

4 One useful property of a Hough transform is that the pixels that define a
5 straight line need not all be contiguous in the image space. This property is very
6 useful when detecting lines with short breaks in them due to noise, or when
7 objects are partially occluded by other objects.

8 For example, Fig. 3 is a block diagram that shows that all pixels that lie on
9 a line need not all be contiguous to one another to be categorized as belonging to
10 the same line by a Hough transform. Block 302 is partially occluded by block
11 304. This is evident by the pixels that comprise line segments 306-1 and 306-3. A
12 Hough transform will detect line 306 as comprising each of the following line
13 segments: line segment 306-1, dotted line segment 306-2, and line segment 306-3.
14 Thus, this property of the Hough transform is very useful when detecting lines
15 with short breaks in them due to noise, or when objects are partially occluded by
16 other objects.

17 However, this property also provides misleading results when objects
18 happen to be aligned on purpose or by chance. This is because one disadvantage
19 of the Hough transform methodology is that it detects infinite lines rather than a
20 finite line segment with two well-defined endpoints.

21 For example, Fig. 4 is a block diagram that illustrates that a Hough
22 transform will erroneously detect an infinite line rather than a finite line with well
23 defined endpoints if two or more objects are aligned. Block 402 is aligned with
24 block 404. With respect a particular one top boundary of the two aligned objects,
25 a Hough transform will indicate that there is a single line 406 that includes the

1 following line segments: dotted line segment 406-1, dotted 406-2, and dotted line
2 segment 406-3. However, the particular one top boundary includes two well
3 defined line segments that include: dotted line segment 406-1 belonging to a
4 boundary of block 402, and dotted line segment 406-3 belonging to a boundary of
5 block 404. Indeed, this shows that one disadvantage of the Hough transform
6 methodology is that it detects infinite lines rather than a finite line segment with
7 two well-defined endpoints.

8 This misleading property is especially important to consider, because it
9 substantially complicates the determination of which line segments define
10 boundaries for which objects. When scanning more than one photograph at a time,
11 a user will typically align multiple photographs at a same angle with respect to one
12 another on a scanner platen. This means that two or more objects that share the
13 same orientation will typically be represented as the same peak in the Hough
14 transform space. Thus, multiple scanned objects in image data may be detected as
15 sharing an infinite line, and therefore difficult to classify.

16 In light of the above, two objects that are aligned based on the same
17 orientation can have fewer than eight (8) peaks in the Hough transform space
18 because well-defined aligned lines segments are typically detected as single
19 infinite lines. As discussed above in reference to Fig. 2, a single rectangular object
20 will typically produce four (4) peaks in the Hough space. Thus, two aligned
21 objects may be erroneously classified as a single large object. Moreover, three
22 such aligned objects can have fewer than twelve (12) such peaks, and the like.

23 Thus, the determination of which of the detected straight lines correspond
24 to which objects is prone to substantial error for the following reasons: (a) the
25 generated edge map may either missing data or may include data that do not

1 correspond to straight lines; (b) if the edge map is noisy, any straight lines that are
2 detected may be erroneously segmented due to missing data; and (c) two or more
3 aligned objects may be detected as sharing infinite lines and thereby classified as a
4 single large object. To account for such substantial error, a number of rules are
5 applied with respect to the straight lines that are determined using the Hough
6 transform of the edge map.

7 Before describing the rules in more detail, first note that a user will
8 typically scan in similarly sized objects. Moreover, note that a corollary to the
9 problem of detecting well-defined line segments with respect to aligned objects, is
10 that it is not as difficult to identify which line segments belong to which objects
11 when the each of the objects are oriented at unique angles with respect to one
12 another. This is because such non-aligned objects have distinct and well-defined
13 sets of four peaks in the Hough transform space.

14 With respect to the set of rules that are applied to determine which of the
15 detected lines belong to which of the one or more objects, lines that are parallel
16 and perpendicular to one another are identified as described above in reference to
17 Fig. 2. Lines that are parallel and perpendicular with respect to one another have
18 peaks in (rho, theta) space with theta coordinates that are respectively zero (0)
19 degrees and ninety (90) degrees apart from one another. Identifying lines with
20 parallel and perpendicular orientations is especially useful because one
21 implementation locates rectangular objects such as photographs in the scanned
22 image data. Moreover, the respective dimensions of any detected objects are used
23 to determine the respective dimensions of any other objects that are not so simply
24 detected because of any confusion as to which lines belong to which object(s).

After determining straight lines by taking a Hough transform of the edge map, if any objects are detected, same sized candidate objects with dimensions that are similar to the detected objects are then sought. This correlates with an assumption that a user will typically scan in similar sized objects. A pair of parallel lines that are a distance d_1 apart, where d_1 is one of the dimensions of a previously located object, is considered to identify respective boundaries of a candidate object. If another pair of parallel lines, perpendicular to the first pair, are located that are distance d_2 apart, where d_2 is the other dimension of the previously located object, all four boundaries of the candidate object are considered to have been identified.

In one implementation, upon identifying a line that is a potential boundary of a candidate object after a first object has already been identified, additional checking is performed to determine that the potential boundary does not belong to a feature of the previously identified object. For example an object feature includes a line that includes the image of the object such as a line that separates a sky from a horizon. The additional checking determines if pixel values of the potential boundary and pixel values near the potential boundary coincide with image to background transitions beyond the range of the already identified object. If so, the potential boundary line is considered to represent a valid object boundary line. Otherwise, the line is no longer considered to represent a valid object boundary, but rather a feature of the already identified object. Using this process, the lines of a candidate object are identified.

After lines outlining a candidate object have been identified, it is determined whether the pixel values inside of the candidate object represent substantially image data values and whether the pixel values on the exterior of the

1 candidate object represent the determined background color values. As described,
2 such background data values may correspond to the scanner platen background
3 values. If so, the lines are considered to represent a valid, or “found” object.
4 Otherwise, the lines are not considered to represent a valid object.

5 In one embodiment, objects are classified in an easy to hard progression.
6 Any line which has a single companion parallel line and a single pair of
7 perpendicular lines is a good candidate for an object at an angle distinct from the
8 other objects in the image. If this candidate object satisfies the tests of having
9 substantially non-background interior pixels it is classified as an object, and its
10 dimensions are recorded. This technique typically results in the classification of
11 one or more objects. Next, lines which have more than a single companion
12 parallel line, and/or more than a single pair of companion perpendicular lines are
13 examined. Generally the techniques proceed from lines that represented the larger
14 peaks in the Hough transform space to those that represented smaller peaks.

15 For a given line, a search is made among lines parallel to the given line for
16 lines at a distance that correspond to either of the dimensions of any of the
17 previously found objects. If the search succeeds, a search is performed among
18 lines perpendicular to the given line for pairs of lines that correspond to the other
19 dimension of any of the previously found objects. If a candidate set of four lines
20 is identified that match the dimensions of any previously found object, and if the
21 candidate lines pass the test of having substantially non-background interior
22 pixels, the candidate lines are classified as representing an object. The four lines
23 that bound this object are not deleted from the list of lines—as some of them may
24 also bound other objects.

In this manner, objects that are at distinct angles with respect to one another are identified. Moreover, objects that have the same dimensions as the identified objects at distinct angles are identified. If at the end of this process there are a number of lines that have not been classified as belonging to objects, pairs of the non-classified parallel and perpendicular lines are evaluated to determine if they represent an object with a non-background interior and negligible overlap with the previously found objects.

Exemplary Computer

Fig. 5. shows a computer 102 that forms a suitable environment for the described system 100 of Fig. 1. The components shown in Fig. 5 are only examples, and are not intended to suggest any limitation as to the scope of the functionality of the invention; the invention is not necessarily dependent on the features shown in Fig. 5.

Generally, various different general purpose or special purpose computing system configurations can be used. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The functionality of the computers is embodied in many cases by computer-executable instructions, such as program modules, that are executed by the computers. Generally, program modules include routines, programs, objects,

1 components, data structures, etc. that perform particular tasks or implement
2 particular abstract data types. Tasks might also be performed by remote
3 processing devices that are linked through a communications network. In a
4 distributed computing environment, program modules may be located in both local
5 and remote computer storage media.

6 The instructions and/or program modules are stored at different times in the
7 various computer-readable media that are either part of the computer or that can be
8 read by the computer. Programs are typically distributed, for example, on floppy
9 disks, CD-ROMs, DVD, or some form of communication media such as a
10 modulated signal. From there, they are installed or loaded into the secondary
11 memory of a computer. At execution, they are loaded at least partially into the
12 computer's primary electronic memory. The invention described herein includes
13 these and other various types of computer-readable media when such media
14 contain instructions programs, and/or modules for implementing the steps
15 described below in conjunction with a microprocessor or other data processors.
16 The invention also includes the computer itself when programmed according to
17 the methods and techniques described below.

18 For purposes of illustration, programs and other executable program
19 components such as the operating system are illustrated herein as discrete blocks,
20 although it is recognized that such programs and components reside at various
21 times in different storage components of the computer, and are executed by the
22 data processor(s) of the computer.

23 With reference to Fig. 5, the components of computer 102 may include, but
24 are not limited to, a processing unit 120, a system memory 106, and a system
25 bus 121 that couples various system components including the system memory to

1 the processing unit 120. The system bus 121 may be any of several types of bus
2 structures including a memory bus or memory controller, a peripheral bus, and a
3 local bus using any of a variety of bus architectures. By way of example, and not
4 limitation, such architectures include Industry Standard Architecture (ISA) bus,
5 Micro Channel Architecture (MCA) bus, Enhanced ISA (EISAA) bus, Video
6 Electronics Standards Association (VESA) local bus, and Peripheral Component
7 Interconnect (PCI) bus also known as the Mezzanine bus.

8 Computer 102 typically includes a variety of computer-readable media.
9 Computer-readable media can be any available media that can be accessed by
10 computer 102 and includes both volatile and nonvolatile media, removable and
11 non-removable media. By way of example, and not limitation, computer-readable
12 media may include computer storage media and communication media. Computer
13 storage media includes volatile and nonvolatile, removable and non-removable
14 media implemented in any method or technology for storage of information such
15 as computer-readable instructions, data structures, program modules, or other data.
16 Computer storage media includes, but is not limited to, RAM, ROM, EEPROM,
17 flash memory or other memory technology, CD-ROM, digital versatile disks
18 (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic
19 disk storage or other magnetic storage devices, or any other medium which can be
20 used to store the desired information and which can be accessed by computer 110.
21 Communication media typically embodies computer-readable instructions, data
22 structures, program modules or other data in a modulated data signal such as a
23 carrier wave or other transport mechanism and includes any information delivery
24 media. The term “modulated data signal” means a signal that has one or more if
25 its characteristics set or changed in such a manner as to encode information in the

1 signal. By way of example, and not limitation, communication media includes
2 wired media such as a wired network or direct-wired connection and wireless
3 media such as acoustic, RF, infrared and other wireless media. Combinations of
4 any of the above should also be included within the scope of computer readable
5 media.

6 The system memory 106 includes computer storage media in the form of
7 volatile and/or nonvolatile memory such as read only memory (ROM) 131 and
8 random access memory (RAM) 132. A basic input/output system 133 (BIOS),
9 containing the basic routines that help to transfer information between elements
10 within computer 102, such as during start-up, is typically stored in ROM 131.
11 RAM 132 typically contains data and/or program modules that are immediately
12 accessible to and/or presently being operated on by processing unit 120. By way
13 of example, and not limitation, Fig. 5 illustrates operating system 108, application
14 programs 110, other program modules 136, and program data 112. As is well
15 known, operating systems provide such low-level functions as module (process
16 and dynamic link library) management, scheduling, interprocess messaging,
17 memory management, file system management, and graphical user interface
18 support.

19 The computer 102 may also include other removable/non-removable,
20 volatile/nonvolatile computer storage media. By way of example only, Fig. 5
21 illustrates a hard disk drive 141 that reads from or writes to non-removable,
22 nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to
23 a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that
24 reads from or writes to a removable, nonvolatile optical disk 156 such as a CD
25 ROM or other optical media. Other removable/non-removable,

1 volatile/nonvolatile computer storage media that can be used in the exemplary
2 operating environment include, but are not limited to, magnetic tape cassettes,
3 flash memory cards, digital versatile disks, digital video tape, solid state RAM,
4 solid state ROM, and the like. The hard disk drive 141 is typically connected to
5 the system bus 121 through an non-removable memory interface such as
6 interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically
7 connected to the system bus 121 by a removable memory interface such as
8 interface 150.

9 The drives and their associated computer storage media discussed above
10 and illustrated in Fig. 5 provide storage of computer-readable instructions, data
11 structures, program modules, and other data for computer 102. In Fig. 5, for
12 example, hard disk drive 141 is illustrated as storing operating system 144,
13 application programs 145, other program modules 146, and program data 147.
14 Note that these components can either be the same as or different from operating
15 system 108, application programs 110, other program modules 136, and program
16 data 112. Operating system 144, application programs 145, other program
17 modules 146, and program data 147 are given different numbers here to illustrate
18 that, at a minimum, they are different copies. A user may enter commands and
19 information into the computer 102 through input devices such as a keyboard 162
20 and pointing device 161, commonly referred to as a mouse, trackball, or touch
21 pad. Other input devices (not shown) may include a microphone, joystick, game
22 pad, satellite dish, scanner, or the like. These and other input devices are often
23 connected to the processing unit 120 through a user input interface 160 that is
24 coupled to the system bus, but may be connected by other interface and bus
25 structures, such as a parallel port, game port, or a universal serial bus (USB). A

1 monitor 191 or other type of display device is also connected to the system
2 bus 121 via an interface, such as a video interface 190.

3 The computer may operate in a networked environment using logical
4 connections to one or more remote computers, such as a remote computer 180.
5 The remote computer 180 may be a personal computer, a server, a router, a
6 network PC, a peer device or other common network node, and typically includes
7 many or all of the elements described above relative to computer 102, although
8 only a memory storage device 181 has been illustrated in Fig. 5. The logical
9 connections depicted in Fig. 5 include a local area network (LAN) 171 and a wide
10 area network (WAN) 173, but may also include other networks. Such networking
11 environments are commonplace in offices, enterprise-wide computer networks,
12 intranets, and the Internet.

13 When used in a LAN networking environment, the computer 102 is
14 connected to the LAN 171 through a network interface or adapter 170. When used
15 in a WAN networking environment, the computer 102 typically includes a
16 modem 172 or other means for establishing communications over the WAN 173,
17 such as the Internet. The modem 172, which may be internal or external, may be
18 connected to the system bus 121 via the user input interface 160, or other
19 appropriate mechanism. In a networked environment, program modules depicted
20 relative to the computer 102, or portions thereof, may be stored in the remote
21 memory storage device. By way of example, and not limitation, Fig. 5 illustrates
22 remote application programs 185 as residing on memory device 181. It will be
23 appreciated that the network connections shown are exemplary and other means of
24 establishing a communications link between the computers may be used.

1 **Exemplary Procedure**

2 Fig. 6 illustrates an exemplary procedure 600 to detect the boundaries of
3 multiple objects in image data. At block 602, the procedure obtains the image
4 data. Step 604 generates an edge map from the image data. At block 606, the
5 procedure takes a Hough transform of the edge map to produce a set of domain
6 peaks, each domain peak corresponds to a straight line of an object in the image
7 data. At block 608, the procedure identifies a first object with a first dimension
8 based on the set of domain peaks.

9 At block 610, the procedure determines if there are any line pairs
10 remaining. At block 612, the procedure evaluates the remaining lines to identify
11 any other objects in the image. This is accomplished by identifying (a) line sets
12 that indicate an object at a distinct angle as compared to an orientation of a
13 previously found object (e.g., block 608), (b) lines sets that indicate an object
14 having a same dimension as a previously found object, and (c) pairs of parallel and
15 perpendicular lines indicating an object that satisfies a substantially non-
16 background interior condition and a negligible overlap condition with a previously
17 found object.

18 Fig. 7 is a block diagram that shows further aspects of an exemplary
19 procedure 700 to detect multiple objects in image data. Specifically, Fig. 7
20 illustrates more details of block 612 if Fig. 6. At block 702, the procedure
21 determines if there is a candidate object(s) having four (4) lines and dimensions
22 that correspond to a previously found object (e.g., see, block 608, Fig. 6). If so, at
23 block 704, the procedure classifies the candidate object(s) as a found object by
24 adding it to a list of objects identified in the image data.

At block 706, the procedure 700 determines if there is a candidate object having three (3) lines and dimensions that correspond to a previously found object (e.g., see, block 608, Fig. 6, and block 704). If so, at block 708, the procedure determines if the candidate object(s) satisfy the image interior and background color constraints that are described in greater detail above. If so, at block 710, the procedure classifies the candidate object(s) as a found object by adding it to a list of objects identified in the image data.

At block 712, the procedure 700 determines if there is a candidate object having three (3) lines that is at an angle that is distinct from a previously found object (e.g., see, block 608, Fig. 6, block 704, and block 710). If so, at block 714, the procedure determines if the candidate object(s) satisfy the image interior and background color constraints that are described in greater detail above. If so, at block 716, the procedure classifies the candidate object(s) as a found object by adding it to a list of objects identified in the image data. In this manner the procedure evaluates a candidate object to determine if the candidate object represents an object in the image.

Alternative Embodiments

The Hough transform is computationally expensive with respect to both the amount of processing time and processing power that is required to analyze image data to detect straight lines. There is a possibility that a user will only scan a single object. While the above described procedure will determine that there is only a single object represented by image data by locating only four peaks in the Hough transform space, a less computationally expensive way to determine that the image data only represents a single object is desired. Thus, one

1 implementation avoids such expensive processor computations by providing a
2 lightweight procedure to first determine whether more than a single object is
3 represented by image data without needing to transform the edge map data into
4 Hough space.

5 To accomplish this, transitions between the background and the image data
6 and between the image data and the background are determined while generating
7 the edge map. On the I-th row, the first transition between the background and the
8 image data is called *left(i)*. The last transition between the image data and the
9 background is called *right(i)*. On the J-th column, the first transition from top to
10 bottom between the background and the image data is called *top(j)*. The last
11 transition between the image data and the background data is called *bottom(j)*.
12 Respective histograms that represent the differences between *right(i)* minus *left(i)*
13 and *top(j)* minus *bottom(j)* are generated.

14 Next, the generated histograms are analyzed to identify certain
15 characteristics such as well-defined peaks or clusters of data. A determination is
16 made as to whether the image data includes one or more objects based on these
17 identified characteristics.

18 For example:

- 19 • If the histograms include only a single peak, the image data represents only a
20 single object. (These dimensions may not be perfectly horizontal and/or
21 vertical).
- 22 • If the histograms include only two peaks, the image data represents multiple
23 objects.

- 1 • If there is a gap in either the first histogram or the second histogram, then the
2 image data represents multiple objects. A gap is the absence of color transition
3 in either the horizontal or vertical dimension.
4 • If neither (a), (b), or (c) apply, the image data represents multiple objects.

5 The above described criteria, or “rules” provide a way to make a rapid
6 determination of whether image data includes more than a single object without
7 transforming the edge map into Hough transform space. These rules are
8 conservative in that they will almost never result in image data that includes
9 multiple objects being classified as including only a single object, while
10 occasionally classifying such image data as containing a single object. This is
11 considered to be a good tradeoff because the consequences of misclassification in
12 the former case are that a user must then manually designate the objects to be
13 segmented. Whereas, misclassification in the latter case is merely the expenditure
14 of some excess computation (such computation is represented by blocks 610
15 and 612 of Fig. 6, and blocks 702-716 of Fig. 7) to determine which line segments
16 belong to which object(s).

17 In this manner, inventive subject matter detects whether image data such as
18 scanned image data includes multiple objects. Regardless of whether one object
19 or more than one object in the image data is detected, the described systems and
20 procedures determine each respective object boundary. These determined
21 boundaries can be used by image processing software to automatically segment
22 the detected objects from the image data. Thus, a user that places multiple objects
23 onto a scanner platen for scanning may not be required to manually segment the
24 multiple objects from a single larger object.

1 **Conclusion**

2 Traditional image processing techniques typically detect only a single large
3 object upon scanning multiple objects that are placed on a scanner platen. Thus,
4 such traditional systems and procedures do not generally provide for the automatic
5 segmentation of multiple objects from scanned image data. Rather, traditional
6 techniques, after automatically detecting a single object, generally require a user to
7 perform manual segmentation of image data to segment any additional objects
8 from the scanned image data. In contrast to such traditional image processing
9 techniques, the described subject matter automatically detects boundaries of
10 multiple objects in image data, identifies one or more objects based on the
11 detected boundaries, and segments the one or more identified objects from the
12 image data. This is done automatically, without requiring a user to manually
13 segment out the separate objects from the image data by hand.

14 Although the subject matter has been described in language specific to
15 structural features and/or methodological steps, it is to be understood that the
16 subject matter defined in the appended claims is not necessarily limited to the
17 specific features or steps described. Rather, the specific features and steps are
18 disclosed as preferred forms of implementing the claimed subject matter.